# Rule Pruning Techniques in the Ant-Miner Classification Algorithm and Its Variants: A Review

Hayder Naser Khraibet AL-Behadili[1], Ku Ruhana Ku-Mahamud[2], Rafid Sagban[3]
[1]College of Arts and Sciences, Universiti Utara Malaysia, Kedah, Malaysia
[2] Data Science Research Lab, School of Computing, Universiti Utara Malaysia, Kedah, Malaysia
[3] Computer Science Department, University of Babylon, Babylon, Iraq
E-mail: hayder_naser@ahsgs.uum.edu.my, ruhana@uum.edu.my, rsagban@uobabylon.edu.iq

***Abstract*— Rule-based classification is considered an important task of data classification. The ant-mining rule-based classification algorithm, inspired from the ant colony optimization algorithm, shows a comparable performance and outperforms in some application domains to the existing methods in the literature. One problem that often arises in any rule-based classification is the overfitting problem. Rule pruning is a framework to avoid overfitting. Furthermore, we find that the influence of rule pruning in ant-miner classification algorithms is equivalent to that of local search in stochastic methods when they aim to search for more improvement for each candidate solution. In this paper, we review the history of the pruning techniques in ant-miner and its variants. These techniques are classified into post-pruning, pre-pruning and hybrid-pruning. In addition, we compare and analyse the advantages and disadvantages of these methods. Finally, future research direction to find new hybrid rule pruning techniques are provided.**

***Keywords—Rule based classification; Ant colony optiomazation; Rule Induction ; Knowledge discovery.***

## I. Introduction

The gap between the generation of data and its understanding is ever-increasing [1]. Understanding the hidden value in data is necessary to determine its specific characteristics. Data mining (DM) uses machine learning, statistics, and artificial intelligence power to analyze data and discover knowledge [2]. DM is either a prediction or description task. The prediction task is important in various disciplines including physical sciences, social sciences, humanities, medicine, and business to establish a model from available data. The model is used to predict an unknown value of a variable of perspective data. The common type of prediction task is classification. The classification task is the process of producing a model to predict a class of unlabeled instances from input data. In this task, interesting data (or patterns) are extracted from real-world datasets. The classification task involves different ways to represent the knowledge. Common classification techniques include decision tree classifier, neural network technique, support vector machine, k-nearest neighbor classifier, and rule-based classifier [3]. All classifiers in DM aim to introduce high prediction accuracy but focus out in the degree of human comprehensibility. Thus, numerous problems are encountered in the interpretation and explanation of relationships between features. Nevertheless, rule-based classifier is one of the most common, supervised machine learning techniques and is considered an easy decision-making method due to its simplicity, and outstanding performance [4]; furthermore, it is easily applied in any intelligent system [5].Classification in this technique consists of a list of prediction rules as follows:

*If <term1> and <term2> and ..... then <target class>*

The (IF)-part and the (THEN)-part of the rule are called rule antecedent and consequence, respectively. The antecedent consists of one or more terms which are conditions that test the attributes. For example, *blood pressure= high*. On the other hand, the rule consequence represents the prediction class.

Rule-based classification algorithms and techniques are categorized into two main classes [6]. The first class is the divide and conquer approach, in which a variety of algorithms are introduced to translate different classifier techniques into a set of rules such as decision tree classifier. The second class is the separate and conquer approach, which can directly generate *If - Then* rules from a dataset using different rule-based algorithms such as ant-miner [7]. Furthermore, experiments showed that the ant-miner algorithm achieves similar performance to other classification techniques, and it outperforms in some application domains [8]–[10].

One problem that often arises in any rule-based classification is the overfitting problem, in which the prediction rules are complex and consist of a large number of terms [11]. The rule will present a perfect fit (high predictive accuracy) for specific instances from which they are generated, but generalizing them to a new dataset is difficult. The accuracy rate of predicting rules in unseen instances would be seriously affected. Therefore, accuracy and complexity are the main challenges for any rule-based classification. Rule pruning is a framework used to avoid overfitting. Post-pruning, pre-pruning, and hybrid pruning techniques are used in ant-miner studies to remove terms that do not contribute to correct prediction [9]. Given its characteristics, rule pruning plays an important role in constructing a classification model in ant-miner and its variants.

In this paper, we discuss the overfitting problem and present a comprehensible review on rule pruning techniques used in the ant-miner classification algorithm and its variants to overcome the above-mentioned problem.

The rest of this paper is organized as follows. In section 2, we describe the ant colony optimization algorithm and its application in different NP-hard problems. The descriptions of the ant-miner algorithm and its components are provided in Section 3. In Section 4, we discuss issues concerning overfitting and underfitting. We also provide a taxonomy of pruning techniques and describe the methods used in the rule pruning in ant-miner and its variants. In Section 5, we highlight the strengths and weaknesses of each technique and provide future research directions to overcome the drawbacks of the original pruning procedure. Finally, Section 6 concludes the research.

## II. Ant Colony Optimization

Ant colony optimization (ACO) is a population-based metaheuristic algorithm for optimization problems. It initially appeared at the beginning of 1990s by Dorigo [12]. ACO is inspired by the behavior of real ants to find the shortest path between a food source and the nest, despite being almost blind. Initially, every ant searches randomly for a food nest and deposits its own chemical trail called a pheromone whenever it travels. This pheromone acts as an indirect communication mechanism among the ants. Together with many ants searching the paths, the overall paths are affected by the pheromone substance laid by those ants. The pheromone concentration builds in the path that is selected by more ants (short path) and increases its probability to be chosen. By contrast, the pheromone intensity on the long path evaporates and disappears with time. In this way, the algorithm uses the characteristic of individuals cooperating to adopt the stochastic decision-making policy based on local information. Furthermore, the ACO algorithm has been successfully applied in many NP-hard combinatorial optimization problems, such as the classic travelling salesman problem, graph coloring, job shop, project scheduling, multiple knapsack, connection-oriented network routing, and DM classification task [13].

## III. Ant-Miner algorithm

In the field of DM rule-based classification, the first proposed system using the ACO algorithm is ant-miner. This method, which was proposed by Parpinelli, Lopes, and Freitas (2002), is inspired from the foraging behavior of ant colonies in the real world. It uses stochastic behavior and memory to provide a predicting rule list that is completely understandable.

The ant-miner algorithm expands as a swarm-based, separate and conquer, metaheuristic, and stochastic approach. It starts with all training data instances to discover one classification rule. A rule is then added to discover a rule list in which each instance satisfies this rule antecedent (if-part) and the rule consequence (then-part) is removed from the training set. This process will stop when the instances in the dataset are less than pre-specified threshold values, known as Max_uncovered_cases. This process consists of three main procedures, namely, rule construction, rule pruning, and pheromone updating.

In rule construction, each ant starts to select terms to be added to the rule antecedent. The rule term is a specific (attribute, value) pair from the dataset, and each attribute can only be used one time under the rule. The ant will add one term that improves the predictive accuracy based on its amount of information and its pheromone intensity until one of the following criteria comes to a stop. In the first criterion, all the attributes are used. In the second, the minimum number of instances is covered by the current rule. Once the rule antecedent is finished, the system selects the rule consequence by assigning the majority class among the cases covered by the rule.

Rule pruning aims to reduce the size of the discovered rules to increase their comprehensibility. It prunes one term at a time while it improves rule quality. The procedure loops until no more improvement occurs, or at least one term is left under the rule, as shown in Fig.1. The class value of the dataset can potentially change during this procedure because the majority classes in the instances covered by the pruned rule might change compared with those covered by the original rule.

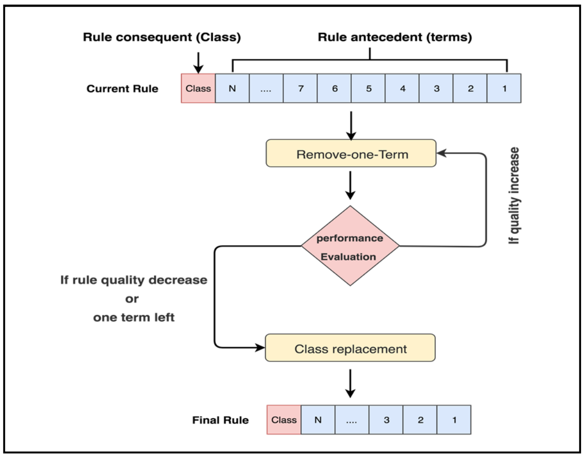

Fig. 1. Rule Pruning in the original ant-miner algorithm

In pheromone updating, after rule construction and rule pruning, the pheromone is updated. Pheromone updating involves two basic steps. First, increasing the amount of the pheromone in all terms appears in the rule based on rule quality. Second, evaporating every term does not occur in the rule. It is achieved by normalizing the unused terms. Another ant will then build its rule by benefitting from the updated amount of pheromone. The process is completed until one of the following stopping criteria is met. For the first criterion, the number of ants should be equal to or greater than the number of discover rules. The second criterion depends on the rule convergence threshold, at which the ant starts to converge by constructing a rule similar to the one constructed before. The best rule among all construction rules will be added to the list of discover rules. Subsequently, the algorithm will begin a new loop by initially setting all terms with the same pheromone amount.

## IV. Rule Pruning

Rule pruning is a common technique in rule-based classifiers that reduces the size of the discovered rules by avoiding the overfitting noisy data [14]. Noise in a dataset is caused by certain reasons (incorrect input, programming errors, and hardware failures). Such noisy data have a detrimental effect by misguiding the learning algorithm and producing a very poor classifier performance. In the learning process, the algorithm adds terms to the rule to increase its predictive accuracy by fitting the instances too closely. In this way, the rule will cover positive instances (instances correctly predicted by the rule) and then remove them from the training set; thereafter, the new instances are subjected to a new rule generation step. Subsequently, this type of rule is a perfect fit for instances from which they were generated. By contrast, when the rules are generated from noisy instances, they are highly complicated, lack usefulness, and exhibit low predictive accuracy on classifying unseen instances. This problem is known as overfitting, which can occur when the constructed rules fit too well, or exactly, to a particular training instance and do not have the applicability to unseen data. Then, those rules negatively influence the whole performance of the training model. An example of the overfitting rule picked up from the ant-miner algorithm without using rule pruning occurs in an experiment undertaken on a breast cancer dataset from the Ljubljana UCI Machine Learning Repository and is shown in Fig. 2. This dataset consists of nine attributes, all of which appear in the rule. Moreover, it can be observed that the rule is a perfect fit to the data instances from where it is generated.

*IF age ='50-59' AND menopause ='ge40' AND tumor-size ='20-24' AND inv-nodes ='3-5' AND node-caps= 'yes' AND deg-malig ='2' AND breast ='right' AND breast-quad ='left_up' AND irradiat ='no' THEN 'no-recurrence-events'*

Fig. 2. Example of overfitting rule

The abovementioned problem can be solved by detecting the significant terms in the generation rule and pruning the irrelative terms that provide minimal quality to classify the instances. This mechanism aims to improve the accuracy of the rule and increase its simplicity [15]. Post-pruning, pre-pruning, and hybrid-pruning are three strategies used in the ant-miner rule-based classifier. In the pre-pruning strategy, the rule discovery algorithm is halted before creating a full rule. The stopping condition handles irrelevant terms during the learning process (i.e., stop selecting the term when the impurity measured for some terms is less than the pre-deterministic value). However, post-pruning deals with irrelevant terms after an overfitting rule has been constructed; in this strategy, the rule grows to maximum size. Then, the irrelative term is deleted from the rule. Meanwhile, hybrid-pruning combines the characteristics of post-pruning and pre-pruning. In order to observe the influence of rule pruning, an experiment was undertaken with the same Ljubljana breast cancer dataset, using the same parameter setting on the ant-miner algorithm with the traditional post-pruning procedure. Then, we obtained a different rule from similar instances as shown in Fig. 3. This rule involved only two attributes in its structure. Thus, the rule is simpler and has less number of terms. Conversely, it covers more instances and is more accurate.

*IF menopause = 'ge40' AND irradiat = 'no' THEN 'no-recurrence-events'*

Fig. 3. Example of pruning rule

The above examples are provided to show the impact of rule pruning in ant-miner classification algorithms which is equivalent to that of local search in stochastic methods. The pruning procedure aims to search for an improvement for each candidate solution produced by each ant. In addition, it increases its simplicity. In contrast, designing the algorithm without pruning techniques as in the MACO algorithm [16] will introduce complex rules and may face overfitting problems. Given its characteristics, rule pruning plays an important role to construct a classification model in ant-miner and its variants. In addition, in all pruning strategies, any excess of pre-pruning and post-pruning in the rule may lead to a very simple rule that does not have the ability to capture the underlying structure of the data. This problem is known as underfitting. Thus, the rule will not be suitable and lead to poor predictive performance on the data. Fig. 4 shows the overfitting and underfitting rules based on predictive error and model complexity.

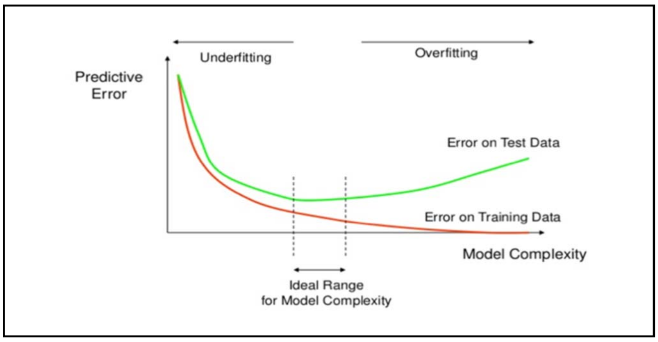


Fig. 4. Overfitting and underfitting effect on error

### A. Post-pruning technique in ant-miner variants

The traditional technique used in the majority of ant-miner variants is the post-pruning technique. This pruning procedure, used in the majority of algorithms, was inspired from the method proposed in [17]. It removes one term at a time from the rule, thereby improving the quality of the rule. This procedure iterates until no further improvement occurs, or only one term is left in the rule. The post-pruning procedure is then terminated to avoid the underfitting rule. The class value of the dataset can potentially change during this procedure, because the majority of classes in the instances covered by the pruned rule might be changed compared with those covered by original rules [9, 18–32]. Furthermore, the algorithms proposed by [33–38] still use the same traditional procedure to prune the rule, but they introduce a new fitness function to test quality. In addition, a dubbed threshold-aware pruning mechanism and new fitness function are used in algorithms [39, 40] sensible to the order of terms that contain consistent continuous values. This mechanism removes the last term added to the rule for simplification until the rule quality decreases when the last term is removed or the rule has only one term left. Another algorithm provides a new fitness measurement function and simplifies the traditional prune procedure by allowing the consequence part of the rule to remain unchanged during the pruning process [33].

A new post-pruning method was introduced to only prune the best rule discovered by all ants instead of pruning each rule constructed by each ant. Furthermore, the rule quality functions have been changed by using new functions for those algorithms [41–44].

The algorithm that deals with hierarchical multi-label classification [45] considers the replacement of the consequence rule during each single pruning. The pruning procedure removes one term and re-calculates the resulting class because the set of covered instances may change after the pruning of the last term. This step removes one term and replaces the class consequence. It then measures the quality of the candidate rule and compares it with the original rule. If the candidate rule has a higher quality than the original rule, then the former replaces the original rule. This procedure is iterated until no further quality improvement occurs or only one term is left in the rule. In addition, this method uses a distance-based measure as a quality function.

### B. *Pre-pruning technique in ant-miner variants*

The pre-pruning criteria [46, 47] in the construction step accepts or rejects a term to be added to the rule based on its strength or importance rather than the post-pruning method. This step will reduce the number of irrelative terms in the rule. However, this criterion, based on threshold value, aims to reduce the complexity of post-pruning by disallowing the irrelative term to be part of discovered rules.

The algorithm [44] removes the pruning procedure and introduces a new mechanism that extends the domain of each attribute with a dummy value “any”. In the rule construction stage, the selection term of “any” value means that a term is not present in the rule antecedent, leading to a shorter rule. In the abovementioned mechanisms, the removal of the post-pruning procedure will make the algorithm less expensive.

### C. *Hybrid pruning technique in ant-miner variants*

Rule pruning is sensitive to the number of attributes in the dataset being mined because a large number of terms will be included in the constructed rule during the construction stage, followed by a large number of iterations of rule pruning to check rule quality during pruning. A new hybrid rule pruning technique was proposed by Chan and Freitas in 2006, which combines both traditional rule pruning based on the rule quality with a new procedure based on information gain. This hybrid pruner depends on a threshold user pre-defined value called r, which represents the total number of terms in the constructed rule. If the number of terms in the constructed rule does not exceed the value of r, then the traditional rule pruning procedure is directly applied. However, if the constructed rule exceeds the value of r, then the procedure first reduces the number of terms in the constructed rule to r value by selecting the number of terms using the pre-calculated term’s information gain. Subsequently, the value of heuristic function is computed in the rule construction stage with respect to specific class. The procedure will select r number of terms using the well-known roulette wheel selection technique, and the newly selected rule is placed straight into the traditional rule pruning procedure [48].

The hybrid pruning procedure proposed by the MuLAM algorithm [49] is computationally expensive with multi class attributes’ replacement. Hence, the pruning step is modified in two aspects to work with this problem. First, it applies a pre-pruning criterion to accept or reject the class attribute to be added to the current rule. This criterion is based on Cramer’s V coefficient. The criterion consists of pre-defined threshold values [50], which adds the class attributes with the largest frequency among all examples covered by the rule. Second, the class attributes remain the same during pruning. Finally, the new rule undergoes the traditional post-pruning procedure. Fig. 5 shows the distribution of the literature (by year of publication) and provides a first attempt for unifying rule pruning techniques based on the pruning component of ant-mining classifiers.

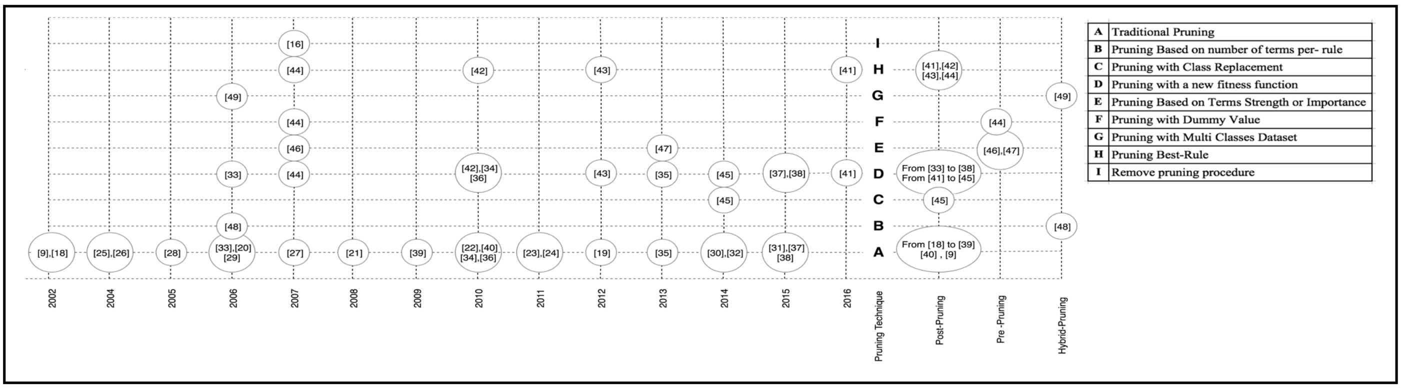

Fig. 5. Methods used in pruning techniques of ant-miner variants

In this study, we provide a year-wise distribution over 35 primary studies of ant-mining algorithms from 2002 to 2016. The bubble at the intersection of axes presents the reference number of primary study on ant-mining algorithm while the right quadrant shows the categories of rule pruning techniques. We observed that the usage of the traditional post-pruning procedure that is produced by the original ant-miner overshadows other methods.

Experiments have been conducted on the literature using three methods in Fig.5 to observe the influence of the rule pruning procedure [9, 46, 48]. Datasets from the UCI Machine Learning Repository have been used by those studies and the overall comparison results show that using the ant-miner algorithm without pruning procedure leads to a decrease in the classification accuracy and produces a very complex rule. Thus, the pruning procedure is emphasized as an important component of any ant-miner algorithm. Overall, the results show that the pre-pruning technique performs better than traditional post-pruning and hybrid pruning in terms of prediction accuracy and simplicity.

## V. DISCUSSION AND FUTURE RESEARCH

Rule pruning has the advantage of selecting subsets of terms from the set of terms in the rule. This type of procedure can simplify rules and make them easier to interpret, increase generalization by decreasing overfitting, and improve predictive accuracy. However, the disadvantage of pre-pruning methods is selecting the right threshold to eliminate the irrelevant terms. If a high threshold value is selected, the discovered rule will underfit the data. By contrast, a low threshold will not overcome the problem of overfitting. The threshold value is considered critical and dependent on the data, and selection of an inappropriate value fails to overcome the problems of overfitting and underfitting. In addition, the extension of the domain of each attribute with dummy values, which was proposed in the literature, will not guarantee that the constructed rule does not consist of any irrelative terms. However, post-pruning is considered complex, time consuming, and costly when a dataset consists of a large number of attributes. In addition, using pruning with class replacement increases the complexity of the procedure. Conversely, simplifying post-pruning by only pruning the best rule discovered by all ants is not ideal because the pruning procedure is equivalent to that of local search in stochastic methods. In this case, although the pruning procedure is run one at a time for the best rule, it will not find high quality rules if it does not explore all rules, or at least an elite set of rules.

Furthermore, fitness or quality function is an important indicator of how close a given construction rule is to achieve a set of objectives to proceed for pruning. An example of objective function is Sensitivity * Specificity defined as:

$$Q=(TP/TP+FN)*(TN/FP+TN) \qquad (1)$$

Where,

TP True positives, the number of cases covered by the rule that have the class predicted by the rule.

FP False positives, the number of cases covered by the rule that have a class different from the class predicted by the rule.

FN False negatives, the number of cases that are not covered by the rule but that have the class predicted by the rule.

TN True negatives, the number of cases that are not covered by the rule and that do not have the class predicted by the rule.

Pheromones are then updated accordingly. The majority of ant-miner variants use the product of sensitivity and specificity. Other variants of ant-miner introduce new fitness functions, which involve different measurements and strategies. For example, [42–44] the original fitness function is replaced with confidence and coverage of the rule. In fact, the coverage is equivalent to a sensitivity indicator in the original ant-miner algorithm. Then, the main idea is replacing the specificity with confidence in the measurement of the rule quality. Therefore, no fitness function has obtained the best predictive performance on all datasets in the literature of ant-mining algorithms. In addition, each fitness function evaluates the candidate rule with different bias and captures varying aspects. One future research direction is to combine the characteristic of different fitness functions to guide the learning process. In addition, the hybrid pruning method is insufficient when a user has to set the threshold value or the number of r parameters (number of terms per rule). The value of this parameter tends to be very critical and dataset-dependent, and user determination may result in a very small rule that is not considered an intelligent way to overcome the problem of overfitting and underfitting. Table. 1 summarizes the advantages and disadvantages of each pruning method used in ant-miner and its variants.

TABLE I. DISADVANTAGES AND ADVANTAGES OF PRUNING TECHNIQUES IN ANT-MINER AND ITS VARIANTS

| Pruning Technique | Type of pruning | Disadvantages | Advantages |
|---|---|---|---|
| Traditional Pruning | Post-pruning | 1-Very complex procedure<br>2-Time-consuming<br>3-Not ideal for high dimensionality datasets | 1-Avoids overfitting<br>2-Improves rule quality<br>3-Produces simple model |
| Pruning based on number of terms per-rule | Hybrid-pruning | 1-Number of terms per rule very critical and selecting inappropriate value can lead to i) overfitting, ii) underfitting, iii) vulnerable rule set | 1-Selecting the appropriate value of term per rule can lead to i) time-saving, ii) simple model, iii) accurate model |
| Pruning with class replacement | Post-pruning | 1-Very complex procedure<br>2-Time-consuming<br>3-Not ideal for high dimensionality datasets | 1-Avoids overfitting<br>2-Improves rule quality<br>3- Produces simple model |
| Pruning with a new fitness function | Post-pruning | 1-Very complex procedure<br>2-Time-consuming<br>3-Not ideal for high dimensionality datasets | 1-Different bias and captures different aspects<br>2-Avoids overfitting |
| Pruning based on terms strength or importance | Pre-pruning | 1- Term strength or importance very critical and data-dependent and selecting inappropriate values can lead to i) underfitting, ii) overfitting | 1-Time-saving<br>2-Selecting appropriate value of term strength will i) avoid overfitting, ii) produce simple model, iii) produce accurate model |
| Pruning with dummy value | Pre-pruning | 1- Irrelative terms included in the rule<br>2-Produces vulnerable rule set | 1-Time-saving<br>2- Produces simple model |
| Pruning with multi classes dataset | Hybrid-pruning | 1-Very complex procedure<br>2-Time-consuming | 1-Deals with multi class datasets<br>2-Avoids overfitting |

| | | | |
|---|---|---|---|
| | | 3-Not ideal for high dimensionality datasets<br>4-The threshold is dataset-dependent | 3- Produces simple model |
| Pruning best-rule | Post-pruning | 1-Does not explore suitable number of rules<br>2-Produces vulnerable rule set<br>3-Overfitting | 1-Very simple procedure<br>2-Time-saving |
| Remove pruning procedure | Without pruning | 1- Overfitting<br>2-Complex model | 1-Time-saving<br>2-Complexity reduction |

For future research directions, the characteristics of pre-pruning and post-pruning can be combined to produce a new hybrid rule pruning technique that features rule accuracy and comprehensibility. This hybrid method may involve a parameter control mechanism, as used in ACO [51], to select an appropriate pre-pruning threshold value (i.e., the number of terms per rule, the term strength, or importance) by using the feedback from the search itself (i.e., the quality of solution) for adjusting the threshold value. Another research direction is to use a deterministic rule to change the threshold value similar to the manner used in simulating annealing. Those deterministic rules are then used to calculate an optimal schedule to change the threshold value during the construction solution. Post-pruning could be set to test the final rule to determine if further pruning is necessary. Post-pruning can accelerate and reduce the complexity process, whereas pre-pruning will avoid the overfitting terms to be included in the rule. Another future research direction is to test the validity of using post-pruning to prune only elitist rules instead of pruning each rule constructed by each ant. This mechanism aims to learn accurate rules and reduce the computational complexity of the post-pruning procedure

## VI. CONCLUSION

Ant-miner is being increasingly used in classification tasks to understand the hidden value and specific characteristics of data in an easy and understandable manner. The effectiveness of the rule pruning procedure in the ant-miner algorithm and it variants are equivalent to local search in stochastic methods, which aims to search for more improvement for each candidate solution produced by each ant. The problems of overfitting and underfitting are common in classification problems. Pruning procedures can overcome those problems by detecting the significant terms in the rule and pruning the irrelative terms that provide minimal quality to classify the instances. The most popular rule pruning techniques used in ant-mining classification algorithms are the traditional post-pruning technique and post-pruning with a new fitness function. The utmost advantage of the pruning technique is improving the quality of the candidate solutions while complexity and time consumption seem to be disadvantages. Overall, the results have been shown that the pre-pruning technique performs better than traditional post-pruning and hybrid pruning in terms of prediction accuracy and simplicity. This review has opened up many more enhancement possibilities by realizing the importance and drawbacks of each method. Our review concludes that drawbacks of current pruning methods can be overcome by combining the good characteristics of pre-pruning and post-pruning via an intelligent design.

## ACKNOWLEDGMENTS

The authors thank the Ministry of Higher Education Malaysia for funding this study under Fundamental Research Grant Scheme (S/O code 13240)

## REFERENCE

[1] X. Wu, X. Zhu, G. Q. Wu, and W. Ding, "Data mining with big data," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 1, pp. 97–107, 2014.

[2] H. N. K. Al-behadili, "Classification Algorithms for Determining Handwritten Digit," *Iraq J. Electr. Electron. Eng.*, vol. 12, no. 1, pp. 96–102, 2016.

[3] I. E. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques*, no. San Francisco, CA. Elsevier, 2005.

[4] J. Cepukenas, C. Lin, and D. Sleeman, "Applying Rule Extraction & Rule Refinement techniques to ( Blackbox ) Classifiers," in *8th International Conference on Knowledge Capture. ACM*, 2015.

[5] H. N. K. Al-behadili, "Intelligent Hypothermia Care System using Ant Colony Optimization for Rules Prediction," *J. babylon Univ.*, vol. 26, no. 2, pp. 47–56, 2016.

[6] A. M. AlMana and M. S. Aksoy, "An Overview of Inductive Learning Algorithms," *Int. J. Comput. Appl.*, vol. 88, no. 4, pp. 20–28, 2014.

[7] H. Elgibreen and M. S. Aksoy, "Rules - TL: A simple and improved rules algorithm for incomplete and large data," *J. Theor. Appl. Inf. Technol.*, vol. 47, no. 1, pp. 28–40, 2013.

[8] J. Uthayakumar, T. Vengattaraman, and P. Dhavachelvan, "Swarm intelligence based classification rule induction (CRI) framework for qualitative and quantitative approach: An application of bankruptcy prediction and credit risk analysis," *J. King Saud Univ. - Comput. Inf. Sci.*, 2017.

[9] R. Parpinelli, H. Lopes, and A. A. AFreitas, "Data Mining With an Ant Colony Optimization Algorithm," *IEEE Trans. Evol. Comput.*, vol. 47, no. 6 (4), pp. 321–332, 2002.

[10] Wu, "Ant colony optimization on building an online delayed diagnosis detection support system for emergency department," in *The 7th International Conference on Computational Intelligence in Economics and Finance (CIEF 2008)*, 2008, pp. 1–7.

[11] L. Yang, K. Li, W. Zhang, and Z. Ke, "Ant colony classification mining algorithm based on pheromone attraction and exclusion," *Soft Comput.*, pp. 1–13, 2016.

[12] M. Dorigo and G. A. Di Caro, "Ant colony optimization : A new meta-heuristic Ant Colon," in *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on. Vol. 2.*, 1999, no. February.

[13] M. Dorigo and T. Stützle, *Ant Colony Optimization*. Cambridge, MA, USA: MIT Press, 2004.

[14] M. Bramer, "An information-theoretic approach to the pre-pruning of classification rules," *Intell. Inf. Process.*, pp. 201–212, 2002.

[15] J. Fürnkranz, "Pruning Algorithms for Rule Learning," *Mach. Learn.*, vol. 27, no. 2, pp. 139–172, 1997.

[16] J. He, D. Long, and C. Chen, "An improved ant-based classifier for intrusion detection," *Third Int. Conf. Nat. Comput. ICNC 2007*, vol. 4, no. 60273062, pp. 819–823, 2007.

[17] J. R. Quinlan, "Generating production rules from decision trees," *Proc. Tenth Int. Jt. Conf. Artif. Intell.*, vol. 30107, pp. 304–307, 1987.

[18] Liu, H. A. Abbass, and B. Mckay, "Density-Based Heuristic for Rule Discovery With Ant-Miner," *6th Australas. Jt. Work. Intell. Evol. Syst.*, pp. 1–6, 2002.

[19] Wu and K. Sun, "A simple heuristic for classification with ant-miner using a population," in *the 2012 4th International Conference on Intelligent Human-Machine Systems and Cybernetics, IHMSC 2012*, 2012, vol. 1, pp. 239–244.

[20] P. Jin, Y. Zhu, K. Hu, and S. Li, "Classification rule mining based on

ant colony optimization algorithm," *Intell. Control Autom.*, no. 1, pp. 654–663, 2006.

[21] F. E. B. Otero, A. A. Freitas, and C. G. Johnson, "cAnt-miner: An ant colony classification algorithm to cope with continuous attributes," *Ant colony Optim. swarm Intell.*, vol. 5217 LNCS, pp. 48–59, 2008.

[22] S. Madhusudhanan, M. Karnan, and K. Rajivgandhi, "Fuzzy based ant miner algorithm in datamining for hepatitis," *2010 Int. Conf. Signal Acquis. Process. ICSAP 2010*, pp. 229–232, 2010.

[23] S. C. Liang, Y. C. Lee, and P. C. Lee, "The application of ant colony optimization to the classification rule problem," *IEEE Int. Conf. Granul. Comput. GrC 2011*, pp. 390–392, 2011.

[24] R. Saian and K. R. Ku-Mahamud, "Hybrid Ant Colony Optimization and Simulated Annealing for Rule Induction," *2011 UKSim 5th Eur. Symp. Comput. Model. Simul.*, no. March 2016, pp. 70–75, 2011.

[25] Liu, H. A. Abbass, and B. Mckay, "Classification rule discovery with ant colony optimization algorithm," in *IEEE Computational Intelligence Bulletin*, 2004, vol. 7435 LNCS, no. 1, pp. 678–687.

[26] Wang and B. Feng, "Classification Rule Mining with an Improved Ant Colony Algorithm," in *Australian conference on artificial intelligence*, 2004, pp. 357–367.

[27] P. Jaganathan, K. Thangavel, A. Pethalakshmi, and M. Karnan, "Classification rule discovery with ant colony optimization and improved quick reduct algorithm," *IAENG Int. J. Comput. Sci.*, no. February, pp. 286–291, 2007.

[28] Jiang, Y. Xu, and Y. Xu, "A novel data mining algorithm based on ant colony system," in *Fourth International Conference on Machine Learning and Cybernetics,* 2005, no. August, pp. 18–21.

[29] J. Ji, N. Zhang, C. Liu, and N. Zhong, "An Ant Colony Optimization Algorithm for Learning Classification Rules," *WI '06 Proc. 2006 IEEE/WIC/ACM Int. Conf. Web Intell.*, pp. 1034–1037, 2006.

[30] B. Fakhar, "A Novel Method for Extracting Classification Rules based on Ant-Miner," *J. Math. Comput. Sci.*, vol. 8, pp. 377–386, 2014.

[31] S. Hota, S. Pranati, and A. K. Jagadev, "Modified Ant Colony Optimization Algorithm (MAnt-Miner) for Classification Rule Mining," in *Advances in Intelligent Systems and Computing*, 2015, vol. 308 AISC, no. VOLUME 1, pp. 267–275.

[32] S. Rajpiplawala and D. K. Singh, "Enhanced cAntMiner PB Algorithm for Induction of Classification Rules using Ant Colony Approach," *IOSR J. Comput. Eng.*, vol. 16, no. 3, pp. 63–72, 2014.

[33] J. Smaldon, "Data mining with an ant colony optimization algorithm," 2006.

[34] K. M. Salama and A. M. Abdelbar, "Extensions to the Ant-Miner classification rule discovery algorithm," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6234 LNCS, pp. 167–178, 2010.

[35] K. M. Salama and F. E. B. Otero, "Using a unified measure function for heuristics, discretization, and rule quality evaluation in Ant-Miner," *2013 IEEE Congr. Evol. Comput. CEC 2013*, pp. 900–907, 2013.

[36] D. Agravat, U. Vaishnav, and P. B. Swadas, "Modified Ant Miner for intrusion detection," in *ICMLC 2010 - The 2nd International Conference on Machine Learning and Computing*, 2010, pp. 228–232.

[37] L. Yang, K. S. Li, W. S. Zhang, Y. Wang, and Z. X. Ke, "Application of a New mAnt-MinerPR Algorithm in Classification Rule Mining," *J. Digit. Inf. Manag.*, vol. 13, no. 5, 2015.

[38] R. Robu, S. Holban, and N. Robu, "Ant Miner Extensions for Improved Data Classification," *Int. J. Comput. Eng. Appl.*, vol. IX, no. Vii, pp. 135–154, 2015.

[39] F. E. B. Otero, A. A. Freitas, and C. G. Johnson, "Handling continuous attributes in ant colony classification algorithms," in *2009 IEEE Symposium on Computational Intelligence and Data Mining, CIDM 2009 - Proceedings*, 2009, pp. 225–231.

[40] I. Michelakos, E. Papageorgiou, and M. Vasilakopoulos, "A hybrid classification algorithm evaluated on medical data," *Proc. Work. Enabling Technol. Infrastruct. Collab. Enterp. WETICE*, pp. 98–103, 2010.

[41] Z. Liang, J. Sun, Q. Lin, Z. Du, J. Chen, and Z. Ming, "A novel multiple rule sets data classification algorithm based on ant colony algorithm," *Appl. Soft Comput.*, vol. 38, pp. 1000–1011, 2016.

[42] W. Shahzad and A. R. Baig, "Compatibility As a Heuristic for Construction of Rules By Artificial Ants," *J. Circuits, Syst. Comput.*, vol. 19, no. 1, pp. 297–306, 2010.

[43] A. R. Baig and W. Shahzad, "A correlation-based ant miner for classification rule discovery," *Neural Comput. Appl.*, vol. 21, no. 2, pp. 219–235, 2012.

[44] D. Martens, M. De Backer, R. Haesen, J. Vanthienen, M. Snoeck, and B. Baesens, "Classification With Ant Colony Optimization," *IEEE Trans. Evol. Comput.*, vol. 11, no. 5, pp. 651–665, 2007.

[45] G. M. Prabha and E. Balraj, "A HM Ant Miner Using Evolutionary Algorithm," *Int. J. Innov. Res. Sci. Eng. Technol.*, vol. 3, no. 3, pp. 1687–1692, 2014.

[46] K. Thangavel and P. Jaganathan, "Rule Mining Algorithm with a New Ant Colony Optimization Algorithm," *Int. Conf. Comput. Intell. Multimed. Appl. ICCIMA 2007*, vol. 1, pp. 561–563, 2007.

[47] S. Tripathy, S. Hota, and P. Satapathy, "MTACO-Miner : Modified Threshold Ant Colony Optimization Miner for Classification Rule Mining," in *Emerging Reserch in Computing, Information, Communication and Application*, 2013, no. August 2013, pp. 1–6.

[48] A. Chan and A. Freitas, "A new classification-rule pruning procedure for an Ant Colony Algorithm," in *International Conference on Artificial Evolution (Evolution Artificielle).*, 2006, vol. 3871 LNCS, pp. 25–36.

[49] A. Chan and A. a. Freitas, "A new ant colony algorithm for multi-label classification with applications in bioinfomatics," in *the 8th annual conference on Genetic and evolutionary computation*, 2006, pp. 27–34.

[50] E. Suzuki, M. Gotoh, and Y. Choki, "Bloomy Decision Tree for Multi-objective Classification," *Princ. Data Min. Knowl. Discov.*, pp. 436–447, 2001.

[51] R. Sagban, K. Ku-Mahamud, and M. Shahbani Abu Bakar, "Nature-inspired Parameter Controllers for ACO-based Reactive Search," *Res. J. Appl. Sci. Eng. Technol.*, vol. 11, no. 1, pp. 109–117, 2015.